

Article

Sustainable Resource Allocation and Reduce Latency Based on Federated-Learning-Enabled Digital Twin in IoT Devices

Mohammed A. Alhartomi ^{1,*}, Adeeb Salh ^{2,*}, Lukman Audah ³, Saeed Alzahrani ¹, Ahmed Alzahmi ¹,
Mohammad R. Altmania ¹, Abdulaziz Alotaibi ⁴, Ruwaybih Alsulami ⁵ and Omar Al-Hartomy ⁶

¹ Department of Electrical Engineering, University of Tabuk, Tabuk 71491, Saudi Arabia

² Faculty of Information and Communication Technology, University Tunku Abdul Rahman (UTAR),
Kampar 31900, Malaysia

³ Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia,
Batu Pahat 86400, Malaysia

⁴ Department of Industrial Engineering, University of Tabuk, Tabuk 71491, Saudi Arabia

⁵ Department of Electrical Engineering, Umm Al-Qura University Makkah, Mecca 24382, Saudi Arabia

⁶ Department of Physics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia

* Correspondence: malhartomi@ut.edu.sa (M.A.A.); adebali@utar.edu.my (A.S.)

Abstract: In this article, we utilize Digital Twins (DT) with edge networks using blockchain technology for reliable real-time data processing and provide a secure, scalable solution to bridge the gap between physical edge networks and digital systems. Then, we suggest a Federated Learning (FL) framework for collaborative computing that runs on a blockchain and is powered by the DT edge network. This framework increases data privacy while enhancing system security and reliability. The provision of sustainable Resource Allocation (RA) and ensure real-time data-processing interaction between Internet of Things (IoT) devices and edge servers depends on a balance between system latency and Energy Consumption (EC) based on the proposed DT-empowered Deep Reinforcement Learning (Deep-RL) agent. The Deep-RL agent evaluates the performance action based on RA actions in DT to distribute its bandwidth resources to IoT devices based on iteration and the actions taken to generate the best policy and enhance learning efficiency at every step. The simulation results show that the proposed Deep-RL-agent-based DT is able to exploit the best policy, select 47.5% of computing activities that are to be carried out locally with 1 MHz bandwidth and minimize the weighted cost of the transmission policy of edge-computing strategies.

Keywords: internet of things; digital twins; resource allocation; energy consumption; deep-RL



Citation: Alhartomi, M.A.; Salh, A.; Audah, L.; Alzahrani, S.; Alzahmi, A.; Altmania, M.R.; Alotaibi, A.; Alsulami, R.; Al-Hartomy, O. Sustainable Resource Allocation and Reduce Latency Based on Federated-Learning-Enabled Digital Twin in IoT Devices. *Sensors* **2023**, *23*, 7262. <https://doi.org/10.3390/s23167262>

Academic Editor: Jose Manuel Molina López

Received: 18 July 2023

Revised: 13 August 2023

Accepted: 16 August 2023

Published: 18 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Beyond Fifth Generation (B5G) architecture with Digital Twin (DT) is anticipated to evolve in the network autonomy and generative intelligence properties, which could optimize and adapt the network. The concept of a DT can be explored to effectively enable the properties of B5G wireless systems. Enhancing the quality of applications and the user experience of services such as autonomous vehicles and smart cities, in practice, depends on evaluating and mining data from the edge network by allocating limited resources and optimizing the network to deliver high-quality services [1,2]. The DT paradigm is one of the most exciting technologies, which can offer instantaneous wireless connectivity and very reliable wireless communication in a B5G network [3]. Due to the huge volume of sent data and the distance between end users and the server, the DT wireless networks are proposed to minimize latency and improve reliability for edge-computing applications [4]. Despite these recent advances and supporting technologies, it is still difficult to create robust control algorithms for physical systems because of the discrepancy between the results of data analysis and the required level of rapid data on their physical systems' conditions [5]. At present, the centralized computing model imposes a significant burden

on communication, which also raises concerns about data security. The Internet of Things (IoT) framework's provision of intelligent services is enabled by data processing based on Artificial Intelligence (AI) algorithms. In IoT networks, the server can gather the operational states and create behavior models for various environmental conditions to create DTs. Solving this problem relies on machine learning techniques being applied to distributed end IoT devices to establish a DT network at the edge. In order to enable secure collaborative learning and develop trust among untrusting users, blockchain-enabled Federated Learning (FL) is capable of improving reliability and enhancing data security in a network. The proposed edge intelligence and a blockchain-powered IoT increase service capacity while lowering edge service costs by utilizing cross-domain sharing and a credit-differentiated transaction approval process [6]. Moreover, the distributed content caching system integrates blockchain technology to improve the efficiency of distributed learning by addressing the problem of data privacy with high-dimensional and time-varying characteristics based on the proposed Deep Reinforcement Learning (Deep-RL) [7].

1.1. Related Works

Building DT models requires the synchronization of a massive amount of data, but the digitization of the IoT is hampered by the limited computing power and communication ability. Our proposed strategy is a framework based on a number of innovative and beneficial technologies and algorithms, such as the incorporation of a blockchain and FL into the DT for edge networks. Many forms of data from a physical entity can be integrated into a digital space thanks to a DT [8]. The efficiency of DT was determined by the analysis of a sizable amount of data created and employed in a variety of applications, such as intelligent scheduling in smart cities, wireless network resources, and industrial real-time monitoring and optimization [9,10]. Some works concentrated on the subject of wireless networks. For instance, a DT edge network was proposed by the authors of [11] to provide a blockchain-enabled Resource Allocation (RA) and scheduling relaying users' manner. To reduce the effects of unreliable communication between users and edge servers, the authors in [12] proposed a DT wireless network to reduce latency and improve reliability for edge-computing applications by mapping IoT devices to DTs in edge servers to improve the efficiency of AI algorithms. In addition, combining the strengths of blockchain and AI has led to significant advances in the delivery of secure and efficient resource management in wireless networks. By integrating AI and blockchain, the authors of [13] suggested a reliable and intelligent resource-sharing system to further build a content caching system using Deep-RL. The authors in [14] proposed a support Vector Machine Training (SVM) technique that protects privacy, called safe SVM, and used blockchain technology to create an IoT network data-sharing framework that is reliable and secure for use by multiple data providers. It is difficult to figure out how to interpret such data using IoT devices with limited resources. The digitalization of the IoT is restricted by the limited computing power and communication abilities, which makes it difficult to adopt blockchain to develop DT models that demand trust and consensus across distributed users. Additionally, blockchain technology enhances distributed learning's effectiveness in solving the problem of data privacy with high-dimensional and time-varying attributes based on the provided Deep-RL.

FL is a new paradigm that is attracting a lot of attention. It offers a brand-new distributed machine learning approach that can reduce the risk of information leaks and hence increase data privacy [15]. As the wireless networks research community's interest in FL grows, a significant amount of effort [16] has been made in this field to enhance FL performance through resource optimization. Based on an empowered FL scheme for DT-edge networks to improve user learning security and data privacy, the authors in [17] presented an FL and permissioned blockchain for DT-edge networks to improve effective communication and information security for IoT applications. To guarantee learning accuracy, reliability, and security, the other authors proposed a dual-driven learning solution for the DT-IoT, enabled by blockchain, to ensure real-time interactions for sustainable computing. This solution depends on combining the DT with an edge network and adopting blockchain

technology [18]. However, the authors of [19] proposed using FL in wireless networks and offered the best compromise between Energy Consumption (EC) and the cost of learning time. In other words, the security of FL has also been discussed in terms of users' data privacy and gradient leakage [20,21]. To increase privacy regarding the loss boundaries in FL and improve security and privacy, the authors in [22] proposed implementing Bayesian differential privacy by an effective method that enables a learning agent to modify its policy and improve data-processing efficiency. The method proposed in [23] applies a new paradigm of DT networks by applying Deep RL. Most recent works [14–16] employ blockchain to build consensus and trust among dispersed users, and guarantee the security and privacy of user data by establishing a distributed ledger amongst users. To fill this gap, this work focuses on achieving real-time data processing and computing to the edge plane, based on the introduction of DT wireless networks, by integrating DTs into wireless networks. Other studies [18–23] did not focus on RA's relationship with user scheduling and bandwidth allocation in IoT devices based on the use of Deep-RL in DTs to evaluate its performance. Moreover, the above studies [18–23] have not investigated sustainable RA or the optimized data relay challenge, wherein Deep Neural Networks (DNN) are harnessed as the strategic schedulers within the strategy scheduler in the recommended solution to balance learning accuracy and time expenditure. The significant contribution of this framework is to establish an enduring RA system, guaranteeing seamless real-time communication between IoT devices and edge servers. This accomplishment hinges on the formulation of an optimized data relay challenge, wherein DNNs are harnessed as the strategic schedulers within the suggested approach. The objective is to strike a harmonious equilibrium between system responsiveness and EC, orchestrated by the innovative utilization of the Deep-RL agent empowered by the proposed DT. The Deep-RL agent evaluates the efficiency of an action by considering RA actions within the DT. According to the iterations and actions taken, this assessment allocates bandwidth resources to IoT devices. The main goal is to come up with an ideal policy that increases learning effectiveness at each stage of the procedure. Then, for collaborative computing, we proposed a blockchain-enabled FL framework running in the DT wireless networks to increase data privacy while enhancing system security and reliability.

1.2. Motivation and Contributions

To increase the IoT devices' performance, and boost the system's reliability, security, and enhance data privacy, we formulated the DT edge networks' model optimization problem and propose a blockchain-enabled FL framework to improve communication efficiency. Accordingly, by simultaneously considering DT association, training data batch size, and bandwidth allotment, we design the data-relaying optimization problem and employ Deep Neural Networks (DNN) as the strategy scheduler in the recommended solution to balance learning accuracy and time expenditure. The main contributions of this paper are as follows:

- To effectively and appropriately optimize IoT networks, we proposed a DT-empowered IoT framework that maps a data-driven DT device's real-time operation using blockchain technology.
- We use the FL framework to build the DT edge network models by employing a gradient descent approach that can lower the overhead of data transfer and safeguard data privacy. Furthermore, we use asynchronous model aggregation to increase communication efficiency, which depends on enhancing the target of local computing by decreasing wait times and keeping track of the training process achievement at edge servers to reduce redundant user delays.
- We present a unique blockchain-supported DT-IoT framework to reduce the system delay and EC and provide secure and reliable computing in DNN, as well as new insight into the impact of the training process achievement requirements on the RA efficiency. The proposed Deep-RL agent based on DT evaluates the performance action based on RA for the user scheduling and bandwidth allocation in IoT devices in order

to increase system stability, develop a balance in learning accuracy, and guarantee the learning accuracy of IoT devices.

2. Materials and Methods

In this study, we present a DT-IoT system that can be integrated with blockchains and users for edge computing. We suggest a blockchain-powered FL architecture to increase security and guarantee the performance of edge computing. It has a user plane and an edge plane that incorporate the DT into the edge network. To achieve secure-aware and reliable-aware edge intelligence, the proposed approach integrates the blockchain and DT based on an increase in the output accuracy and reduction in the loss in terms of DNN in IoT systems. IoT devices receive the aggregated data from DT models following the blockchain consensus process. In order to achieve sustainable RA in the IoT and address the issue of poor accuracy, operational IoT devices can submit data that are sent to edge servers for real-time updates. We set IoT device using the notation $\kappa = \{1, 2, \dots, K\}$. We consider a set κ of K IoT devices in the user layer and connect them to the edge plane via wireless communications. The edge layer is made up of multiple edge servers that have base stations equipped with mobile edge servers. Wireless communications are used by each edge server with the $j = \{1, 2, \dots, J\}$ to communicate with the UEs within its coverage. To ensure secure and reliable data transmission, edge servers are managed as blockchain nodes. To protect real-time IoT devices, edge servers build DTs on the edge plane [18]. The IoT device's data are pre-processed to obtain vectors that reflect the operating state before being used in the real-time twin modeling process. Then, edge servers collect and process IoT device operating states to produce DT models, which are indicated as follows:

$$\mathcal{E} = (p_{\kappa}, R_{\kappa}, D_{\kappa}, F_{\kappa}, \varphi_{\kappa}), \quad (1)$$

where p_{κ} represents the IoT devices' transmission power, R_{κ} represents their upload data rate, D_{κ} represents their pre-processed data set, F_{κ} represents their processing capacity, and φ_{κ} represents the performance index weight parameter of the improvement target in the DT-IoT system.

2.1. Sustainable Blockchain Model for Secure Communication

Data exchange is becoming an essential element of the IoT for DTs and is essential to maintaining IoT security. Reducing the duration of time required for model training in various applications is a crucial challenge in the B5G network due to the expansion of the user devices, the demand for communication with ultra-low latency, and the dynamic network condition. IoT devices communicate data to edge servers in the DT-IoT system enabled by blockchain via orthogonal frequency division multiple access (OFDMA). To transmit data, sub-channels $\mathcal{C}$ are shared with the IoT device κ . The maximal data rate of an IoT device κ is written as

$$R_{\kappa}(\tau) = \sum_{\mathcal{C}=1}^{\mathcal{C}} \beta \zeta_{\kappa, \mathcal{C}} \log_2 \left(\frac{h_{\kappa, \mathcal{C}}(\tau) P_{\kappa, \mathcal{C}}(\tau)}{N_0} \right), \quad (2)$$

where β represents the transmission bandwidth, $h_{\kappa, \mathcal{C}}(\tau)$ represents the IoT device's κ for channel gain at time slot τ , $\zeta_{\kappa, \mathcal{C}}$ represents the number of sub-channels allocated to IoT device κ , $P_{\kappa, \mathcal{C}}(\tau)$ is the IoT device's κ for transmission power in subchannel $\mathcal{C}$, and N_0 is the noise power. The total time delay that an IoT device's DT κ takes to update its status is denoted as

$$\tau_{upd} = \frac{|D_{\kappa}(\tau)|}{R_{\kappa}(\tau)}. \quad (3)$$

The time of data upload depends on a blockchain delay and a communication delay in order to reduce the transmission load as follows:

Firstly: The blockchain can significantly boost the cost-effectiveness of DTs by updating real-time data. Based on an analysis of the consensus procedure of blockchain nodes, the blockchain latency between edge servers consists of the time it takes for information to spread across edge servers and for new blocks to be created $\mathbb{T}_{bd}$.

$$\mathbb{T}_{bd} = \max_j \log_2 \mathcal{N} \left\{ \frac{\mathbb{D}_\kappa \mathbb{T}_j |\omega_e|}{R_{es}} \right\}, \quad (4)$$

where $\mathcal{N}$ represents the number of edge servers, $|\omega_e|$ is the level of transmitted model parameters of κ , and R_{es} represents the achievable data transmission between edge servers.

Secondly: The time for uploaded data depends on communication delays, which depends on the data size of DT $\mathbb{Q}_i$ stored by edge servers, $\int_j^C \mathbb{Q}_i$, which represent the computation speed of the CPU cycle frequency at edge servers j , and $\mathcal{C}_j$ represents the number of CPU cycles consumed to process an edge server to train a data sample. The CPU-cycle frequencies enable operation at various frequencies to regulate power consumption, which helps to shorten the IoT device communication's computation times and speed up the FL training process. The communication delay can be written as

$$\mathbb{T}_{cd} = \max_j \sum_{i=1}^{\mathbb{Q}_j} \frac{\mathbb{D}_{\mathbb{Q}_j}}{\int_j^C \mathbb{Q}_i} \mathcal{C}_j. \quad (5)$$

From the analysis above, the total delay is denoted as

$$\mathbb{T}_{tot} = \mathbb{T}_{bd} + \mathbb{T}_{cd}, \quad (6)$$

where $|\mathbb{D}_\kappa(\tau)|$ represents the data size for IoT device updates.

2.2. Learning Accuracy for DT-IoT-FL

The edge association problem is crucial for minimizing the total time cost in DT edge networks while maintaining the required level of learning accuracy. To evaluate the decision-making capabilities of DTs in our proposed IoT networks, FL is utilized for traffic load prediction and training independent local models based on the locally collected data and then shares their model parameters on the edge servers using wireless transmitting data [17,24]. Blockchain-enabled FL is capable of increasing reliability and enhancing data security in a network to enable secure collaborative learning and foster trust among untrusted users. The proposed method incorporates blockchain and DT to improve output accuracy and lower loss, to create secure-aware and reliable-aware edge intelligence. Due to the proliferation of user devices, it is necessary to shorten the amount of time needed for model training in the various applications, as illustrated in (2) and (3), in order to maintain secure connections. Depending on the particular components of latency being assessed, the dataset utilized for latency calculations in a blockchain setting can change. Reductions in the amount of time needed for model training across a variety of applications due to the growth in user devices and the necessity of ultra-low latency connection are shown in (3). The amount of time it takes to upload data is influenced by the blockchain latency (4) between edge servers and the communication delay for the data size of DT $\mathbb{Q}_i$ that reflects the CPU cycle frequency $\int_j^C \mathbb{Q}_i$ (5). The local dataset of device κ is $\mathbb{D}_\kappa$, which is a collection of data samples with the coordinates $\{\mathbb{Z}_j, \mathbb{Z}_j\}_{j=1}^{\mathbb{Q}_j}$, where $\mathbb{Z}_j$ is sample j -th input and $\mathbb{Z}_j$ is

sample $j - th$ output. Using the data set from IoT device κ , the local loss function can be calculated as

$$F_{\kappa}(\omega) = \frac{1}{D_{\kappa}} \sum_{j=1}^J \sum_{\kappa \in \mathcal{D}_{\kappa}} f_{\kappa}(\omega_j, z_{j\kappa}, z_{j\kappa}), \forall \kappa \in \mathcal{K} \quad (7)$$

where $D_{\kappa} = |\mathcal{D}_{\kappa}|$ represents the number of collected data samples by the IoT device, and $f_{\kappa}(\omega)$ is the loss function that calculates the error in the local training model for data sample j . In order to train model parameters, we employ a gradient descent approach. When iteration begins, edge servers provide the same knowledge to all IoT devices τ . To train the model parameter $\omega(\tau)$, each IoT device integrates its local dataset $\mathcal{D}_{\kappa}$, denoted as:

$$\omega_j(\tau) = \omega(\tau - 1) - \lambda \nabla F_{\kappa} \omega(\tau - 1), \quad (8)$$

where $\lambda > 0$ is the learning step size, and $\nabla F_{\kappa} \omega(\tau - 1)$ indicates the gradient value of the parameter's loss function $\omega(\tau - 1)$. To improve learning accuracy, reducing the EC and URLLC of the DT-IoT system depends on enhancing the target of local computing by decreasing wait times and minimizing the communication load. This depends on keeping track of the training process achievement at edge servers to maintain a global iteration that records the aggregation process' performance in the blockchain. The IoT devices compute updated $\omega_j(\tau)$ in the subsequent iterations by computing the gradient descent of the local loss function in iteration τ . The size of the local training data and the loss value, represented by $\ell = \min_{\kappa \in \mathcal{K}} D_{\kappa} / F_{\kappa}(\omega)$, are used to determine the learning accuracy for each IoT device.

2.3. IoT Device for EC

In this section, local training and data transfer are the two key steps of the EC. The computational EC of CPU frequency generated by IoT devices κ is denoted by f_{κ}^c [20,25]. For local computation, the EC of device κ can be expressed as follows:

$$\epsilon_{\kappa}^c = \delta C_{\kappa} |\mathcal{D}_{\kappa}| \left(f_{\kappa} \right)^2, \quad (9)$$

where C_{κ} is the total amount of processing CPU cycles for IoT devices to train a data sample, $\delta = \gamma \log_2(\frac{1}{\psi})$ represents the effective switched capacitance, γ represents a constant related to the data size and ψ is the minimum loss at this rate. The local model upload requires the following EC:

$$\epsilon_{\kappa} = \frac{p_{\kappa}(\tau) D_{\kappa}(\tau)}{\beta \zeta_{\kappa, C} \log_2 \left(\frac{\hbar_{\kappa, C}(\tau) P_{\kappa, C}(\tau)}{N_0} \right)}. \quad (10)$$

The amount of energy consumed in terms of DT when using an edge server can be written as:

$$\epsilon_j^c = \delta C_j \sum_{i=1}^{Q_j} D_{Q_j} (f_j^c, Q_i)^2. \quad (11)$$

The total EC of the DT-IoT devices connected to FL can be calculated as:

$$\epsilon_{total} = \sum_{\kappa=1}^K (\epsilon_{\kappa}^c + \epsilon_{\kappa}) + \sum_{j=1}^J (\epsilon_j^c). \quad (12)$$

3. Formulation of the Communication Effectiveness Problem for DT-IoT Using FL

In this section, our goal is to develop a federated model that minimizes the weighted cost and enhances learning accuracy from distributed IoT devices. So, the FL model must be trained with minimal resource consumption due to the limited computing and

communication resources of IoT devices. A trade-off between learning accuracy and resource efficiency is established by the combinatorial problem of resource optimization. The optimization problem can be expressed as:

$$\min_{\mathbf{f}, \mathbf{Q}_i, \mathcal{O}} \frac{1}{|\mathcal{D}|} \sum_{j=1}^J \sum_{\kappa=1}^K f_{\kappa}(\omega_j, z_{j\kappa}, z_{j\kappa}) + \sigma(\mathcal{T}_{tot} + \mathcal{E}_{total}) \quad (13)$$

$$\text{s. t. } \mathcal{O} \in \{0, 1\}, \quad (13a)$$

$$\sum_{i=1}^{Q_j} Q_i = K, \quad (13b)$$

$$f_{\kappa}^{min} f_j^{\mathcal{C}} Q_i \leq f_{\kappa} \leq f_j^{max} f_{\kappa}^{max}, \forall \kappa \in K \quad (13c)$$

$$\mathcal{P}_{\kappa}^{min} \leq \mathcal{P}_{\kappa} \leq \mathcal{P}_{\kappa}^{max}, \forall \kappa \in K \quad (13d)$$

$$0 \leq \sigma \leq 1, \quad (13e)$$

where $\mathcal{O}$ is the controlling factor that determines how the loss function and cost functions are balanced, as shown in (13a). The time for uploaded data depends on a number of DT Q_i that represent the IoT device behaviour model created by examining their previous running data during the time slots given to IoT devices at iteration τ by DT $Q_i(\tau)$. The constraint (13c) represents the computing resources allocated for cycle CPU frequency, which are less than the upper or maximum bounds $f_j^{\mathcal{C}}$ and can be solved according to [23]. From (13d), edge servers are shown to have a sufficient power supply compared to IoT devices. From (13e), the scheduling policy vector for IoT devices σ is shown to be able to determine which relay IoT devices will send the nearest training parameter. The transmission time for uploaded data is determined by a number of Q_i that indicate the behaviour model developed by analysing their historical operating data throughout the time slots provided at iteration by $Q_i(\tau)$. The restriction (13c) denotes the computational resources assigned to cycle CPU frequency, which are fewer than the upper or maximum constraints $f_j^{\mathcal{C}}$. In contrast to IoT devices, edge servers from (13d) have a suitable power. As seen in (13c) and (13d), when resources are limited, blockchain technology can be included in both edge and IoT devices. It is difficult to use conventional convex optimization algorithms to successfully solve problems in (13), which is an NP-hard optimization issue. Updates to the global model depend on the minimization of the problem in (13) $\frac{1}{|\mathcal{D}_{\kappa}|} \sum_{j=1}^J \sum_{\kappa \in \mathcal{D}_{\kappa}} f_{\kappa}(\omega_j, z_{j\kappa}, z_{j\kappa})$ according to the gradient descent algorithm used to address $F_{\kappa}(\omega)$, and the ability to obtain the exact learning used on training data and the accuracy of the local data. The objective of communication improvements to edge servers is explained by the equation in the second term (13), as follows:

$$\min_{Q_i, \mathcal{O}} (\mathcal{T}_{tot} + \mathcal{E}_{total}) \quad (14)$$

$$\text{s. t. } \mathcal{O} \in \{0, 1\}, \quad (14a)$$

$$\sum_{i=1}^{Q_j} Q_i = K, \quad (14b)$$

$$\mathcal{T}_{tot} \leq \mathcal{T}_{th}, \quad (14c)$$

$$\mathcal{E}_{total} \leq \mathcal{E}_{th}, \quad (14d)$$

where $\mathcal{T}_{th}$ and $\mathcal{E}_{th}$ represent the estimated threshold for all IoT devices in the DT for delay and EC respectively. As all edge servers have DT models $\mathcal{Q}_i$, the number of IoT devices is equal to the number of edge servers according to (14b). Maintaining DTs and training data in real-time would be difficult because of the predicted growth in IoT devices and data traffic in B5G networks, which means that providing sustainable RA and processing large amounts of data become considerably more complicated. Each device needs to reduce the EC of devices and the costs of system delays. In addition, the efficiency of the learning accuracy and sustainability of the DT-IoT system should be guaranteed by maintaining an appropriate level of URLLC and ensuring the learning accuracy of IoT devices. In order to provide sustainable intelligent contact between IoT devices and edge servers, and to increase the effectiveness of integrated DTs with edge networks, real-time data processing must strike a balance between system delays and EC. To provide sustainable RA, we proposed a DT-empowered Deep-RL-based algorithm to train DNN and generate the policy for each IoT device.

3.1. DNN-Train-Based Resource Scheduling Algorithm in DT-IoT

The DNN can explore the DT to avoid the loss of training caused by exploration-intensive resource consumption in real-time based on transmitting the “straggler’s” transmission tasks to the users with higher connectivity. The model ω_j that was trained by κ has insufficient communication abilities; to improve the communication efficiency, we propose an IoT device $\mathfrak{H}$ that might be used to relay the transmission activities of the model κ . The potential IoT devices $\mathfrak{H}$ can perform all the transmission requirements of κ to determine whether ω_j should be transmitted from κ to $\mathfrak{H}$, and decide on the responsibility of the scheduling policy to obtain the value of $\mathcal{O} \in \{0, 1\}$. After that, IoT devices with the $\mathcal{Q}_i$ designation are provided with bandwidth resources based on their statuses. Using DNN, IoT devices can become more responsive to their surroundings and use less network capacity by uploading fewer pieces of data to edge servers and saving valuable network bandwidth [26,27]. The DNN is represented as $\phi = \{\mathfrak{W}^\ell, \mathfrak{b}^\ell\}$, where $\ell \in \{1, \dots, \mathcal{L}\}$ represents the DNN’s ℓ – th layer. Using the weight vectors $\mathfrak{W}^\ell$ and bias in vector $\mathfrak{b}^\ell$, the DNN can be written as:

$$\mathcal{Z}^\ell = \mathfrak{f}(\mathfrak{W}^\ell \mathcal{Y}^\ell + \mathfrak{b}^\ell). \quad (15)$$

Let $\mathcal{Y}^\ell$ represent the input to the ℓ – th layer of the network, and $\mathcal{Z}^\ell$ is the output of neuron $\mathfrak{n}$ in layer l . The system’s states, such as the transmission rate that can be achieved and the computational power of each IoT device, are input into the DNN during the learning process. The output of the DNN is the best relay policy. The DNN models are trained using the training data in each epoch $(\mathcal{R}, \mathfrak{f}, \mathcal{Q}_i, \mathcal{T})$. In order to reduce the training complexity in DNN, we proposed using a Deep-RL agent to train the policy to reduce the scale of training data, so that they are be smaller in volume and resource-efficient.

3.2. Deep-RL Agent for RA

Reinforcement learning has been successfully used to handle RA and task resource scheduling problems in the DT-IoT. Deep-RL evaluates the performance action based on RA actions $(\mathcal{O}_\tau, \mathcal{Q}_\tau)$, where each agent is optimally assigned in DT to distribute its bandwidth resources to IoT devices based on iteration and the actions taken to distribute resources $(\mathcal{O}_\tau, \mathcal{Q}_\tau)$. Thus, the action-value function $\mathbb{A}$, state-space δ , and reward $\mathfrak{R}$ of the Deep-RL framework are all explicitly defined. Furthermore, the Deep-RL framework takes the

required action $\mathbf{a}_\tau \in \mathbf{A}$, which consists of bandwidth RA $\mathbb{Q}$ and IoT devices scheduling $\mathcal{O}$ at every state $\mathbf{s}_\tau \in \delta$. The state environment can be defined as $\mathbf{s}_\tau = \left\{ F_\tau, f_\tau, R_\tau, \omega_\tau \right\}$, where F_τ represents loss value, f_τ is the speed of the CPU cycle frequency at edge servers, R_τ is the data rate vector that can be achieved and regulated by the bandwidth allocation policy, and ω_τ represents the learning for IoT device [25,28]. To achieve high efficiency in the learning accuracy and sustainability of the DT-IoT, the agent continues to the next state and receives a reward immediately. The reward function $\mathfrak{R}_\tau$ can be defined as:

$$\mathfrak{R}_\tau = \begin{cases} \lambda T_{tot} + C & \text{if } f, P, \varepsilon \leq \varepsilon_{total} \\ \lambda T_{tot} - C & \text{otherwise} \end{cases}, \quad (16)$$

where C represents the number of CPU cycles executed to train data for IoT devices. From (16), a positive reward $\lambda T_{tot} + C$ will be added and encouraged as a suitable trade-off to balance the RA between computing and communication resources if all metrics following the action pass the constraint check. Using a positive reward $\lambda T_{tot} + C$ in combination with an active learning technique can greatly reduce the cost of training a model. Based on the current states obtained from DT, the performance of the action is quantified by the reward function at the end of the iteration. The constraints in (16) if $\varepsilon \leq \varepsilon_{total}$ will also reduce system time costs and every agent attempts to utilize the optimum policy to maximize an accumulative reward based on frequently updated $\mathcal{O}$ and $\mathbb{Q}$, minimizing the weighted cost of the transmission policy and enhancing learning efficiency at every step. Otherwise, when the training sample for CPU C was negative, the iteration completion time and the reward $\mathfrak{R}(\tau)$ have a negative relationship. To evaluate and improve the reward function of in DT-IoT, the future cumulative discounted reward at a time slot can be defined as follows:

$$\mathfrak{r}_{\kappa, \tau} = \mathfrak{R}_{\tau+1} + \varnothing \mathfrak{R}_{\tau+2} + \dots = \sum_{\tau=1}^{T-1} \Gamma^\kappa \mathfrak{R}_{\tau+\kappa+1}, \quad (17)$$

where $\varnothing \in [0, 1]$ represents the discount factor. Based on the appropriate running states from the DT-enabled exploration, the DNN experience training data are obtained $\mathbf{s}_\tau = \left\{ F_\tau, f_\tau, R_\tau, \omega_\tau \right\}$. The final output is obtained by applying the activation function f to Y^ℓ in the hidden layers using the Rectified Linear Unit $ReLU(Y^\ell) = \max(0, Y^\ell)$.

Applying the activation function f to Y^ℓ in the hidden layers using the Rectified Linear Unit $ReLU(Y^\ell) = \max(0, Y^\ell)$ yielded the desired output [29]. Using the states $\mathbf{s}_\tau$ as the input to DNN, the vectors of $\mathcal{O}$ and $\mathbb{Q}$ are the output for the relaying strategies towards network states, which are investigated and generated as:

$$\mathcal{O} = \sum_{\ell=1}^L f\left(\mathfrak{w}^\ell Y^\ell + \mathfrak{b}^\ell\right). \quad (18)$$

Minimizing the weighted cost of transmission policy for distributed IoT devices in (13) depends on adjusting the values of $\mathcal{O}_{j\kappa} \in \mathcal{O}$, assigning the relevant transmission time $\mathbb{Q}_j$ and saving this into replay memory as training samples when $R_\kappa \leq R_j$ and $f_\kappa \geq f_j$.

The minimization of the cost necessary to provide the optimal relaying strategy for state s_τ can be achieved as follows:

$$\begin{aligned} \mathcal{O}_{\kappa j} \max \sum_{\kappa}^K \sum_j^J \mathcal{O}_{j\kappa} \times \mathbb{Q}_j \times \frac{c}{f_{j\kappa} \times R_{j\kappa}} \\ \text{s.t. } \mathcal{O}_{j\kappa} \in \{0, 1\}, \\ \sum_{i=1}^J \mathbb{Q}_i = K, \\ \mathcal{P}_{\kappa} \leq \mathcal{O}_{j\kappa} \times \mathbb{Q}_{j\kappa} \times f_{j\kappa}, \\ \mathcal{P}_{\kappa}^{\min} \leq \mathcal{O}_{j\kappa} \times \mathbb{Q}_{j\kappa} \times f_{j\kappa} \leq \mathcal{P}_{\kappa}^{\max}, \end{aligned} \quad (19)$$

In (19), every agent attempts to utilize the optimum policy to maximize the accumulative reward based on frequently updated $\mathcal{O}$ and $\mathbb{Q}$, minimizing the weighted cost of transmission policy and enhancing learning efficiency at every step, as shown in Algorithm 1 and Figure 1 [12,20].

Algorithm 1: Deep-RL Agent for RA Scheduling Algorithm in DT-IoT

1. Input Set $\mathbb{W}, \mathbb{b}, \mathcal{O}, \mathcal{P}_{\kappa}, f_{\kappa}$ and $\mathbb{Q}$;
 2. Output: value $\mathcal{O}$ satisfying (19);
 3. For each edge server $j \leq J$, complete the following;
 4. Initialize DT-IoT environment setup;
 5. For each time slot τ , complete the following;
 6. Evaluate the performance action based on RA actions $(\mathcal{O}_{\tau}, \mathbb{Q}_{\tau})$;
 7. Check the suitable trade-off to balance the RA between computing and communication according to (16);
 8. End for;
 9. Compute cumulative discounted reward at a time slot and transit to the next state s_τ in DNN based on the actions taken to distribute resources $(\mathcal{O}_{\tau}, \mathbb{Q}_{\tau})$;
 10. Store states and optimal allocation $(\mathcal{O}_{\tau}, \mathbb{Q}_{\tau})$ into replay memory when $R_{\kappa} \leq R_j$ and $f_{\kappa} \geq f_j$;
 11. Minimize the cost necessary to obtain the optimal relaying strategy according to (18) and (19);
 12. End for;
 13. Train the DNN model using the previously saved samples $(s_\tau, \mathcal{O}_{\tau})$.
-

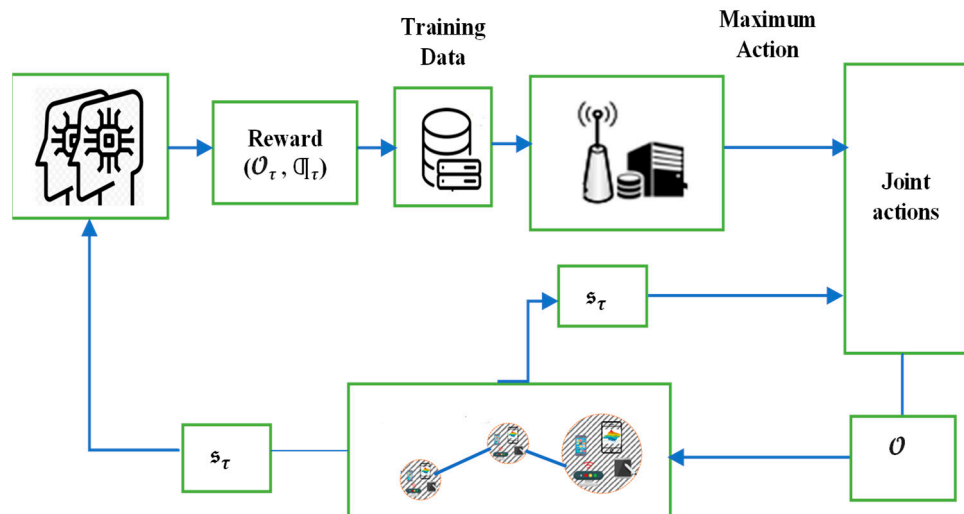


Figure 1. A DT-enabled Deep-RL for edges' association.

4. Discussion

This section in the DT-IoT system proposes the evaluated methods for the Proposed Deep-RL agent based on the DT-RA algorithm. Here, a DT-IoT system with 4 edge servers and 16 IoT devices is built with help from the blockchain. We evenly divide the channel gain ($h_{\kappa, C}(\tau)$) into 11 levels and quantize it as [30] before determining the boundary values. FL efficiency is measured using the MNIST real-world dataset [31] and the Fashion-MNIST dataset [32]. Each dataset has a training set with 60,000 examples and a testing set with 10,000 examples as show in Table 1 [20,26].

Table 1. Simulation parameters.

Parameter	Value
Communication bandwidth	30 MHz
Transmit power $\mathcal{P}$	20 dBm
Adjusting the target weight [1,6,12]	0.8
CPU clock speed mobile edge server [17,19]	1.2×10^{12} Hz
Discount factor [25–27]	0.5
Time slot duration τ [1,3,10]	0.05
Noise figure	5 dB
The data size of each IoT device [25,28]	[0~ 50] MB

In comparison to other methods, the proposed Deep-RL agent based on DT offers the best reward performance, quickest convergence, and most stable learning process. Exploring various data-relaying policies tells the agent how to maximize the overall reward, as shown in (16).

The size of the steps taken to update the model parameters during training is determined by the learning step size, which is an important variable shown in Figure 2. Convergence speed and stability are impacted. A new transaction is added to the blockchain every time a learning step involves changing the learning step size. The iteration number, learning step size, and possibly other pertinent metadata are all recorded in this transaction, along with other pertinent information. The best reward value for user scheduling and bandwidth allocation is attained when the learning accuracy is $\hat{\lambda} = 0.001$, more than when it is $\hat{\lambda} = 0.003$. From Figure 2, the best reward is obtained when the learning strategy maximizes the learning accuracy through more than 4000 iterations. The stringent latency requirements of the increasing number of IoT devices may make delayed convergence impossible to achieve. As training iterations are extended, the worst performance only depends on the immediate reward based on RA for the user scheduling and bandwidth allocation, which increases overall loss.

Figure 3 shows the time cost related to the increasing number of iterations. The discount factor in Deep-RL, as seen in Figure 3, symbolises the preference for future benefits over present ones. Through the blockchain, transactions on a distributed network are verified and agreed upon. Due to the increase in training from $\varnothing = 0.6$ to $\varnothing = 0.8$, this process naturally generates time delays. The blockchain has the ability to contribute value in terms of data openness to avoid limiting the effectiveness of Deep-RL training. It is directly able to reduce the time required to train a Deep-RL agent. When the discount factor is $\varnothing = 0.6$, the time required to run our Deep-RL-based agent is substantially lower; this is because of the process of iterative exploration. Using a positive reward $\hat{\lambda}\mathcal{T}_{tot} + \mathcal{C}$ in combination with an active learning technique can greatly reduce the cost of training a model. This will also reduce system time costs and promote appropriate trade-offs between RA and communication resources. To learn the best policies, the agents must interact with the environment for a long time. The overall time investment for training or testing in each iteration is comparable, despite occasional variations in the outcome curves, as shown in Figure 3. An increased discount factor somewhat results in an increase in time costs in each iteration; this is due to the fact that the policy-training procedure requires more computation when the discount factor is large.

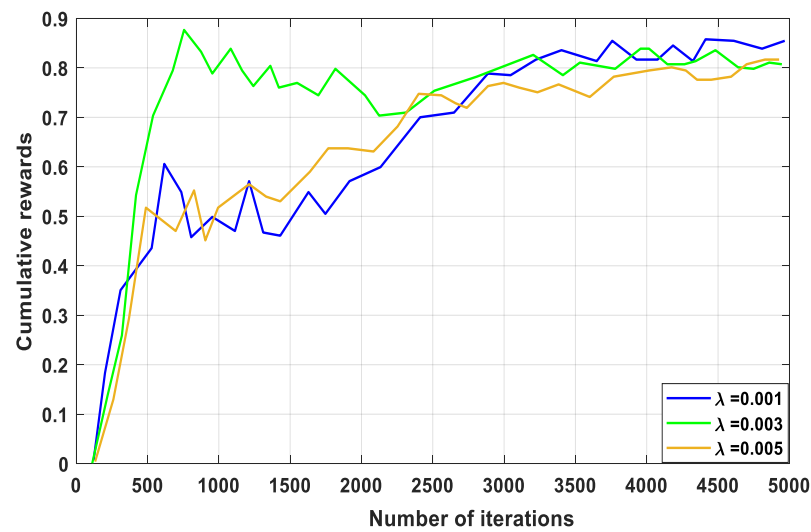


Figure 2. Cumulative reward versus number of iterations.

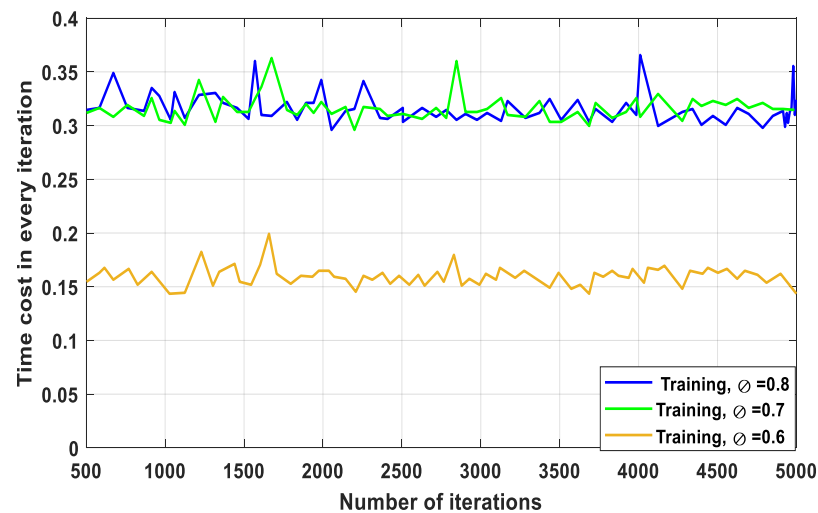


Figure 3. Time cost related to number of iterations.

In Figure 4, the proposed Deep-RL agent based on DT and the learning-agent-based random policy with random edge association are compared in terms of latency performance with respect to training rounds. Comparing the proposed Deep-RL agent to the RA strategy, it is found that the proposed Deep-RL agent greatly lowers the system time cost compared to the RA policy [D. Yueyue 2020]. In order to increase operational efficiency and decrease system latency for our scheme, the Deep-RL agent optimizes DT associations and optimally distributes communication resources. In addition, the learning agent based on random policy leads to a larger time cost compared to the proposed RA policy [D. Yueyue 2020]. The Deep-RL-agent-based DT can produce trained learning accuracy with good performance and use the best policy to maximize an accumulative reward based on an $\mathcal{O}$ that is updated on a regular basis, and $\mathcal{Q}$ that can make predictions on new or unseen data.

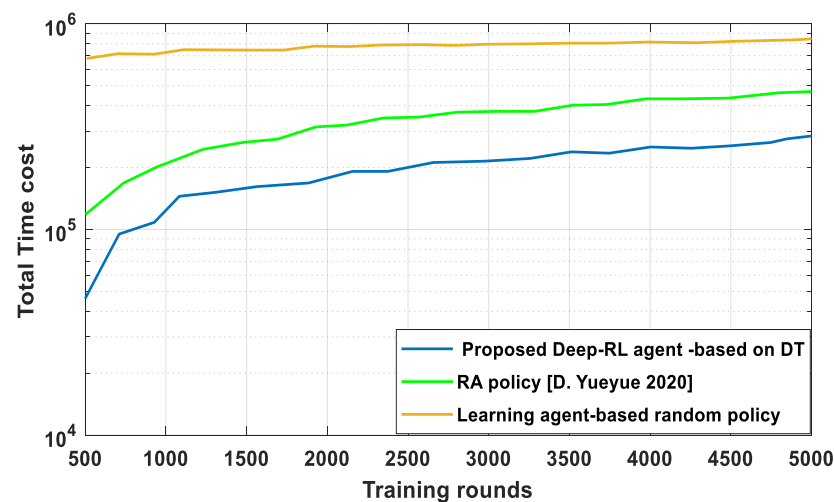


Figure 4. Total time cost versus training round.

Figure 5 shows the highest achievable accuracy of the performance action based on RA actions in DT. Centralized training is the performance upper bound used to set the central trainer's processing power to the total number of devices while ignoring the communication delay associated with collecting training data. The use of DT with Deep-RL agents offers good accuracy. This is because DT ideally assigns each agent a greater level of training to distribute bandwidth resources to IoT devices based on iteration and the actions carried out to distribute optimal resources allocation ($\mathcal{O}_\tau, \mathcal{Q}_\tau$), which had a greater statistical power to more accurately and efficiently allocate resources. According to Figure 5, the proposed Deep-RL agent-based DT trade-off between the number of rounds and the latency per round is able to enhance the scheduling of more IoT devices and lower the number of rounds needed to achieve a fixed accuracy, but at the expense of a higher latency per round. From Figure 5, the latency per round can be decreased by scheduling fewer devices, but the convergence rate with respect to the number of rounds will be slower.

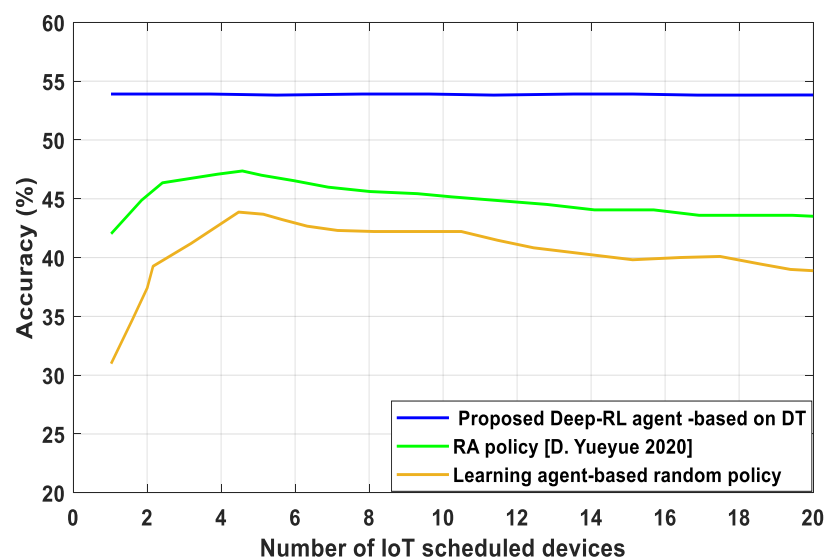


Figure 5. Accuracy versus number of IoT scheduled devices.

Figure 6 shows the total cost of the impact of transmission bandwidth. As bandwidth increases, the total cost declines, and the transmission times are shortened when the communication resource is sufficient. As the overall system bandwidth is increased, it is evident from the simulation results that all curves steadily decline. This is because the increasing

bandwidth allocation has a substantial impact on increasing the data transmission rate between edge servers in DT and IoT devices. With the increased edge server processing capacity, the overall cost of edge computing strategies is greatly reduced. The proposed technique performs well when bandwidth is limited. When compared to the RA policy [D. Yueyue 2020] and the learning-agent-based random policy, respectively, our proposed algorithm achieves time reductions of 1.10 and 1.12, more than those obtained when the system bandwidth is 10 MHz. The proposed Deep-RL-agent-based DT selects 47.5% of computation processes to be carried out locally with 1 MHz bandwidth, compared to the RA strategy [D. Yueyue 2020], which only selects 43%.

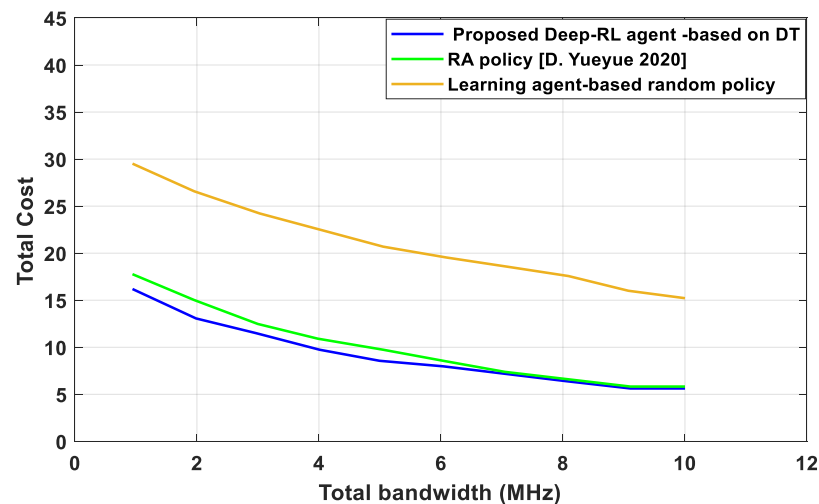


Figure 6. Total cost versus bandwidth.

From Figure 7, the efficiency of the proposed Deep-RL agent based on DT with RA policy [D. Yueyue 2020], the IoT edge resource is allocated based on the average processing demand of two services. When the edge resource is limited, the suggested Deep-RL agent based on DT outperforms the proposed RA policy [D. Yueyue 2020] by a wide margin. The performance advantage in lowering the service latency specifically drops from 1.26 at 1 GHz CPU frequency to only 0.56 at 1.2 GHz CPU frequency. The effect of the number of CPU cycles to be treated depends on the length of local model training, which was dependent on the number of CPU cycles; when many IoT devices are needed to fix the CPU frequency, the energy consumption significantly increases. The explanation is that an efficient RA is more crucial in settings where resources are scarce than it is in scenarios where resources are abundant. The outcomes verify the efficiency of the RA optimization subroutine for edge servers.

A system performance comparison is carried out between the proposed Deep-RL agent based on the DT algorithm and other algorithms to assess the effects of varying IoT device numbers. To confirm the effectiveness of trials, it is important to compare performance under various IoT device counts. Our suggested approach offers a larger reward, as seen in Figure 8. To achieve high-efficiency learning accuracy from (13c), (13d), the total energy cost $\mathcal{E}_{total}$ is also checked as a budget that should not exceed an expected value if $\mathcal{J}, \mathcal{P}, \mathcal{E} \leq \mathcal{E}_{total}$; then, the agent continues on to the next state and receives a reward immediately. In addition, the reward of the Deep-RL agent based on the DT leads to a better performance than the RA policy [D. Yueyue 2020], algorithm and learning-agent-based random policy, which reduces the EC. This is because the Deep-RL agent based on the DT algorithm improves the learning accuracy and takes the required action by solving the problem of a huge action-space. Also, it is clear that the reward increases significantly as the assignment time increases.

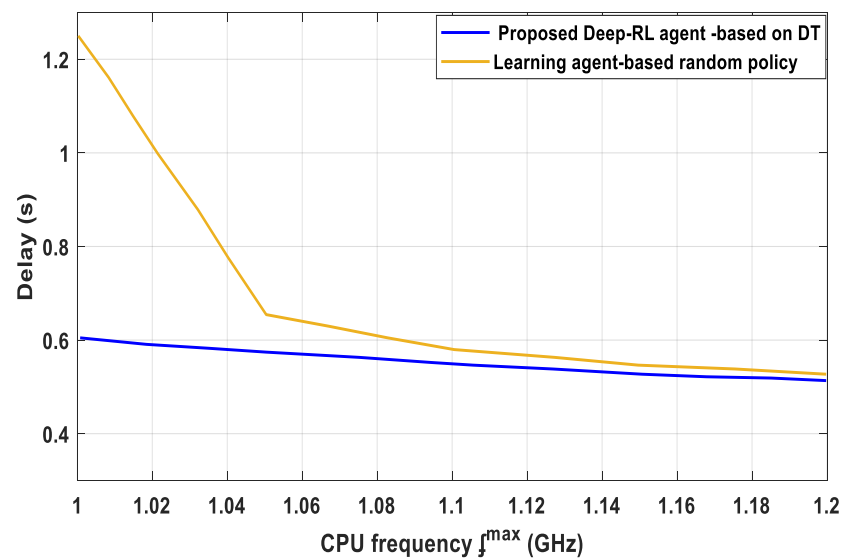


Figure 7. Relay versus CPU frequency.

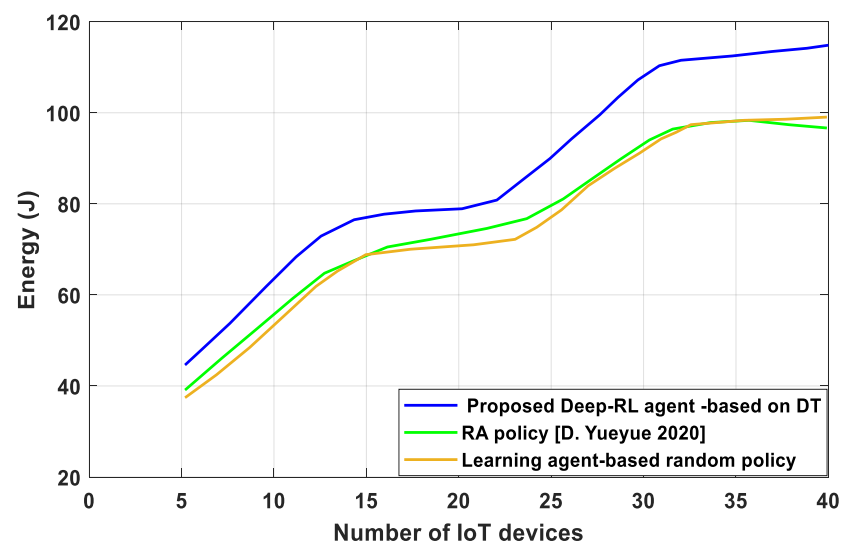


Figure 8. EC versus number of IoT devices.

5. Conclusions

In this paper, we proposed the use of sustainable RA- in FL-DT edge networks, which integrates DT into edge networks for real-time data analysis. We also focused on enhancing the security and effectiveness of edge computing in IoT networks, ensuring sustainable computing and analyzing the limitations of the current RA. The B5G networks rely on improvements in the security and effectiveness of edge computing in IoT networks, which require URLLC, real-time data processing, and real-time data analysis to achieve sustainable computing, and mitigate the computation and communication capacity. Moreover, minimizing the cost of system delay and the EC depends on the proposed DT-enabled Deep-RL agent technique for optimal IoT devices to guarantee the efficiency of the learning accuracy and sustainability of the DT-IoT system FL by employing DNN for RA. The simulation results demonstrate that the proposed Deep-RL-agent-based DT can balance system delay and EC to improve reliability by guaranteeing the learning accuracy of IoT devices. This also greatly lowers the system time cost and offers good accuracy, enhancing the efficiency of the integrated DTs with edge networks.

Author Contributions: Conceptualization, M.A.A., A.S., S.A. and L.A.; methodology, A.S., S.A. and M.A.A.; software, A.S. and M.A.A.; validation, A.S., S.A. and M.A.A.; formal analysis, A.S., A.A. (Abdulaziz Alotaibi), R.A. and O.A.-H.; investigation, S.A. and M.R.A.; resources, A.S. and L.A.; writing—review and editing, A.S., A.A. (Ahmed Alzahmi) and M.A.A.; visualization, A.S.; supervision, M.A.A. and A.S.; funding acquisition, S.A., M.A.A. and O.A.-H. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (S-1443-0183).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data can be obtained from the open access online dataset from MATLAB Math Work: <https://in.mathworks.com/help/deeplearning/gs/create-simple-sequence-classification-network.html>.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (S-1443-0183).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, W.; Xiang, W.; Yang, Y.; Cheng, P. Optimizing federated learning with deep reinforcement learning for digital twin empowered industrial IoT. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1884–1893. [CrossRef]
2. Salh, A.; Audah, L.; Shah, N.S.; Alhamadi, A.; Abdullah, Q.; Kim, Y.H.; Al-Gailani, S.A.; Hamzah, S.A.; Esmail, B.A.; Almohammed, A.A. A survey on deep learning for ultra-reliable and low-latency communications challenges on 6G wireless systems. *IEEE Access* **2021**, *9*, 55098–55131. [CrossRef]
3. Wu, Y.; Zhang, K.; Zhang, Y. Digital twin networks: A survey. *IEEE Internet Things J.* **2021**, *8*, 13789–13804. [CrossRef]
4. Schluse, M.; Priggemeyer, M.; Atorf, L.; Rossmann, J. Experimentable digital twins—Streamlining simulation-based systems engineering for industry 4.0. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1722–1731. [CrossRef]
5. Zhang, K.; Zhu, Y.; Maharjan, S.; Zhang, Y. Edge intelligence and blockchain empowered 5G beyond for the industrial Internet of Things. *IEEE Netw.* **2019**, *33*, 12–19. [CrossRef]
6. Wang, P.; Xu, N.; Sun, W.; Wang, G.; Zhang, Y. Distributed incentives and digital twin for resource allocation in air-assisted internet of vehicles. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, 29 March–1 April 2021; pp. 1–6.
7. Dai, Y.; Xu, D.; Zhang, K.; Maharjan, S.; Zhang, Y. Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4312–4324. [CrossRef]
8. Rasheed, A.; San, O.; Kvamsdal, T. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access* **2020**, *8*, 21980–22012. [CrossRef]
9. Zhang, H.; Zhang, G.; Yan, Q. Dynamic resource allocation optimization for digital twin-driven smart shopfloor. In Proceedings of the 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 27–29 March 2018; pp. 1–5.
10. Sun, W.; Wang, P.; Xu, N.; Wang, G.; Zhang, Y. Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted internet of vehicles. *IEEE Internet Things J.* **2021**, *9*, 5839–5852. [CrossRef]
11. Zhang, K.; Cao, J.; Maharjan, S.; Zhang, Y. Digital twin empowered content caching in social-aware vehicular edge networks. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 239–251. [CrossRef]
12. Razzaq, S.; Shah, B.; Iqbal, F.; Ilyas, M.; Maqbool, F.; Rocha, A. Deep ClassRooms: A deep learning based digital twin framework for on-campus class rooms. *Neural Comput. Appl.* **2023**, *35*, 8017–8026. [CrossRef]
13. Dai, Y.; Xu, D.; Maharjan, S.; Chen, Z.; He, Q.; Zhang, Y. Blockchain and deep reinforcement learning empowered intelligent 5G beyond. *IEEE Netw.* **2019**, *33*, 10–17. [CrossRef]
14. Shen, M.; Tang, X.; Zhu, L.; Du, X.; Guizani, M. Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities. *IEEE Internet Things J.* **2019**, *6*, 7702–7712. [CrossRef]
15. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Blockchain and federated learning for 5G beyond. *IEEE Netw.* **2020**, *35*, 219–225. [CrossRef]
16. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
17. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Communication-efficient federated learning and permissioned blockchain for digital twin edge networks. *IEEE Internet Things J.* **2020**, *8*, 2276–2288. [CrossRef]

18. Wang, D.; Li, B.; Song, B.; Liu, Y.; Muhammad, K.; Zhou, X. Dual-driven resource management for sustainable computing in the blockchain-supported digital twin IoT. *IEEE Internet Things J.* **2022**, *10*, 6549–6560. [\[CrossRef\]](#)
19. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Communication-efficient federated learning for digital twin edge networks in industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5709–5718. [\[CrossRef\]](#)
20. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5098–5107. [\[CrossRef\]](#)
21. Wang, Z.; Song, M.; Zhang, Z.; Song, Y.; Wang, Q.; Qi, H. Beyond inferring class representatives: User-level privacy leakage from federated learning. In Proceedings of the IEEE INFOCOM 2019–IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 2512–2520.
22. Triastcyn, A.; Faltings, B. Federated learning with bayesian differential privacy. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2587–2596.
23. Dai, Y.; Zhang, K.; Maharjan, S.; Zhang, Y. Deep reinforcement learning for stochastic computation offloading in digital twin networks. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4968–4977. [\[CrossRef\]](#)
24. Tran, N.H.; Bao, W.; Zomaya, A.; Nguyen, M.N.; Hong, C.S. Federated learning over wireless networks: Optimization model design and analysis. In Proceedings of the IEEE INFOCOM 2019–IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1387–1395.
25. Salh, A.; Ngah, R.; Audah, L.; Kim, K.S.; Abdullah, Q.; Al-Moliki, Y.M.; Aljaloud, K.A.; Talib, H.N. Energy-Efficient Federated Learning with Resource Allocation for Green IoT Edge Intelligence in B5G. *IEEE Access* **2023**, *11*, 16353–16367. [\[CrossRef\]](#)
26. Wang, H.; Sievert, S.; Liu, S.; Charles, Z.; Papailiopoulos, D.; Wright, S. Atomo: Communication-efficient learning via atomic sparsification. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–11.
27. Caldas, S.; Konečný, J.; McMahan, H.B.; Talwalkar, A. Expanding the reach of federated learning by reducing client resource requirements. *arXiv* **2018**, arXiv:1812.07210.
28. Yuan, N.; He, W.; Shen, J.; Qiu, X.; Guo, S.; Li, W. Delay-aware NFV resource allocation with deep reinforcement learning. In Proceedings of the NOMS 2020–2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–7.
29. Dinh, C.T.; Tran, N.H.; Nguyen, M.N.; Hong, C.S.; Bao, W.; Zomaya, A.Y.; Gramoli, V. Federated learning over wireless networks: Convergence analysis and resource allocation. *IEEE/ACM Trans. Netw.* **2020**, *29*, 398–409. [\[CrossRef\]](#)
30. Wang, D.; Zhao, N.; Song, B.; Lin, P.; Yu, F.R. Resource management for secure computation offloading in softwarized cyber-physical systems. *IEEE Internet Things J.* **2021**, *8*, 9294–9304. [\[CrossRef\]](#)
31. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#)
32. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.